

UsingPhp

Using PHP With Tomcat

This tutorial shows how to use PHP version 4 with Tomcat version 4 or later. The tutorial was originally written and sent to the tomcat-dev mailing list by Jean-Frederic Clere (on his vacation, no less 😊) and Henri Gomez.

PHP version 5 is not currently supported as it does not include the necessary servlet code.

Prerequisites

- Download PHP (this tutorial uses PHP 4.3.5)
- Download Tomcat (this tutorial uses Tomcat 5.0.19)
- Define \$JAVA_HOME for your JDK installation
- Define \$TOMCAT_HOME for your Tomcat installation
- Define \$PHP_HOME for your PHP installation

Patch for PHP configure Script

There is a patch required to compile PHP 4 to use Tomcat 5.

Prior to version 2.4 of Servlet Specification, the name of the servlet jar file was `servlet.jar`. In version 2.4 of the Servlet Specification, this name was changed to `servlet-api.jar`. Tomcat 4 uses the name `servlet.jar`, whereas Tomcat 5 and later uses `servlet-api.jar`. This causes problems with PHP's configure script.

This patch for PHP's configure script will fix this problem:

```
--- configure.org      2004-04-07 11:20:24.000000000 +0200
+++ configure         2004-04-07 11:22:50.000000000 +0200
     if test "$withval" = "yes"; then
         SERVLET_CLASSPATH=.
     else
+    if test -f $withval/common/lib/servlet-api.jar; then
+        SERVLET_CLASSPATH=$withval/common/lib/servlet-api.jar
+    fi
+
     if test -f $withval/lib/servlet.jar; then
         SERVLET_CLASSPATH=$withval/lib/servlet.jar
     fi
```

Patch for sapi/servlet/servlet.java

enum is now a reserved word with Java 5, thus causing `servlet.java` to break the make process.

```
--- servlet.java.orig  2005-09-26 22:25:55.000000000 -0400
+++ servlet.java       2005-09-26 22:26:11.000000000 -0400
@@ -63,12 +63,12 @@
     if (!request.getMethod().equals("POST")) {
         result = request.getQueryString();
     } else {
-        Enumeration enum = request.getParameterNames();
+        Enumeration xenum = request.getParameterNames();
         String concat = "";
         result = "";

-        while (enum.hasMoreElements()) {
-            String name = (String)enum.nextElement();
+        while (xenum.hasMoreElements()) {
+            String name = (String)xenum.nextElement();
             String value = request.getParameter(name);

             try {
```

PHP Installation

- Extract the source code to PHP in a work directory
- Patch if needed (that is, patch if building PHP to run with Tomcat version 5 or later)
- Run `configure`, then make in the top directory of the PHP sources:

```
./configure --with-servlet=$TOMCAT_HOME --with-java=$JAVA_HOME
make
```

- A jar file and dynamic library are produced from the make: `sapi/servlet/phpsrvlt.jar` and `libs/libphp4.so`.
- Copy the jar file to your web application's class repository, or, alternately, to Tomcat's common class repository (as is shown here):

```
cp $PHP_HOME/sapi/servlet/phpsrvlt.jar $TOMCAT_HOME/common/lib
```

- Declare PHP servlet and servlet-mapping in the web applications `web.xml` file, or in Tomcat's shared `web.xml` file:
 - Copy from `$PHP_HOME/sapi/servlet/web.xml` the servlet and servlet-mapping and paste into the file `$TOMCAT_HOME/conf/web.xml`.
- Modify your `LD_LIBRARY_PATH` to include the dynamic library produced in the first step above:

```
LD_LIBRARY_PATH=$PHP_HOME/libs
export LD_LIBRARY_PATH
```

- As an option, you can put `libphp4.so` someplace where java is already looking, any place in `System.getProperty("java.library.path")`, such as any of:

```
/usr/lib/jdk1.5.0_04/jre/lib/i386/client:/usr/lib/jdk1.5.0_04/jre/lib/i386:/usr/lib/jdk1.5.0_04/jre/../../lib/i386
```

Fedora Core 1 Issues with Tomcat 5.5.9, PHP 4.3.11 and jdk1.5.0_03

This may have just been an issue with the particular system I was building, but I was unable to set `$JAVA_HOME`, `$PHP_HOME`, `$TOMCAT_HOME`, or `$LD_LIBRARY_PATH` at the command line. The workaround was to edit `/etc/profile` and add the variables there (i.e., and the line

```
JAVA_HOME=/usr/java/jdk1.5.0_03
```

and add `JAVA_HOME` to the export variables).

- If `make` returns an error where `javac` is not a recognized command, you'll need to patch the Makefile produced by `./configure`. Look for "`&& javac`" and replace it with the full path to `javac` (i.e., "

```
&& /usr/java/jdk1.5.0_03/bin/javac
```

- If `make` returns an error regarding "`enum`" while trying to build `phpsrvlt.jar`, you'll need to edit `$PHP_HOME/sapi/servlet/servlet.java` and replace `enum` with `xenum`.

Start Tomcat

```
$TOMCAT_HOME/bin/startup.sh.
```

Testing

Verify the following is in your webapp's `web.xml` (creates the servlet entries and maps `.php` to that servlet and mentioned in the PHP installation steps above):

```
<servlet>
  <servlet-name>php</servlet-name>
  <servlet-class>net.php.servlet</servlet-class>
</servlet>
<servlet>
  <servlet-name>php-formatter</servlet-name>
  <servlet-class>net.php.formatter</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>php</servlet-name>
  <url-pattern>*.php</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>php-formatter</servlet-name>
  <url-pattern>*.phps</url-pattern>
</servlet-mapping>
```

Verify that phpsvlt.jar is in you WEB-INF/lib directory, or the tomcat common/lib directory (as mentioned above in the PHP installation steps)

Create a file named test.php in the docBase directory of your webapp.

In that file, simply put:

```
<?php phpinfo(); ?>
```

Point your browser at the file by navigating to <http://localhost:8080/test.php>

If everything is working as it should, you will see an informational status page produced by PHP.

[CategoryFAQ](#)